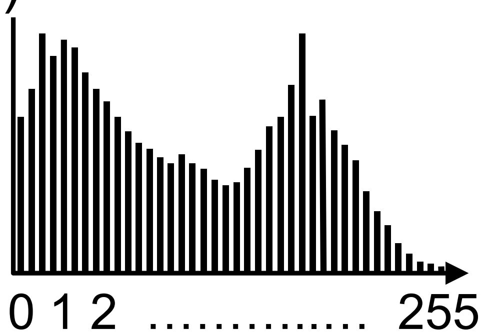
9. 影像區塊分割

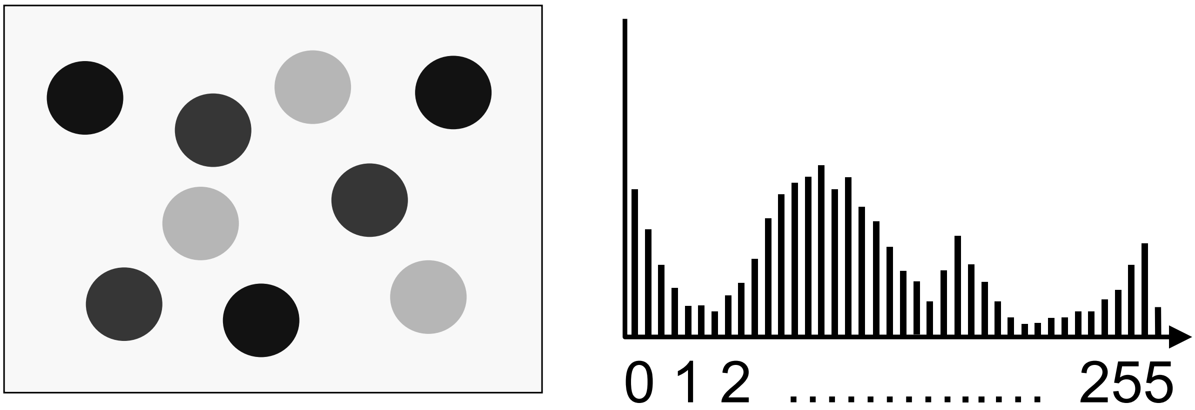
- 影像**區塊分割** (region segmentation) 的意義，是根據像素的相似性 (similarity) 把像素聚集成一塊塊區域 (物以類聚)；也相當於將整張影像分割成一個個區塊。

- 相似性可定義在不同的特徵上；例如，像素的灰階或色彩值、小區塊像素的紋理或幾何結構、區塊的移動或變形、等。我們將區塊分割法分成三類：
  i. 以灰階數量分佈圖為基礎的分割法 (threshold-based segmentation)
  ii. 以區塊為基礎的分割法 (region-based segmentation)
  iii. 以動態資訊為基礎的分割法 (motion-based segmentation)。



- 最基本的區塊分割法就是以灰階或色彩值的不同將像素分成幾類；這類方法稱為灰階數量分佈圖分割法，簡稱分佈圖分割法、長條圖分割法、或門檻值 (閥值) 分割法 (histogram thresholding)，或更簡單稱為**門檻值法、閥值法、閾值法**。

- 這類方法沒有區塊的觀念，只是將像素依照灰階或色彩分成幾類，任何一類都可能包含幾個區塊。



- 以區塊為基礎的分割法是以區塊為出發點，將相似的相鄰像素聚集 (cluster) 成一個個區塊；所以做完這類分割法，區塊就已成形了，不必再做其他後處理。上述多種特徵：像素灰階、色彩值、小區塊像素紋理、幾何結構都可用在這類方法上。這類方法效果比較好，但較複雜，速度也較慢。

- 以動態資訊為基礎的分割法是以影像中物體的移動來分割出該物體區塊，所以要用連續多張影像才能做；這類方法常被用來做視覺監視 (surveillance) 用。

9.1灰階分佈圖門檻值分割法

- 在分析這類方法前，先定義清楚名稱。有三個專有名稱：threshold, threshold value, thresholding
  - Threshold 是動詞，做門檻值法的動作，我們稱為門檻值分割，簡稱**分割**。
  - Threshold value 是用來分群的門檻值，稱為**門檻值**，但許多論文也直接用threshold表示門檻值，這點看上下文是動詞還是名詞就能分辨了
  - Thresholding 是名詞，就是門檻值分割法。例如，”In the thresholding method, we threshold the image with a threshold value *T*.”

- 門檻值法可以作用在任何分佈圖 (histogram) 上；例如，灰階數量分佈圖、色彩值分佈圖、梯度 (gradient) 分佈圖、 .. 等。

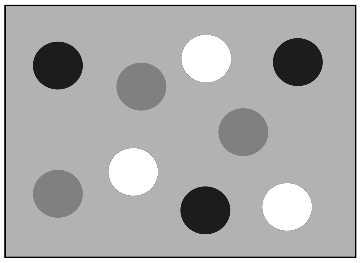
- 門檻值法的目的通常都是要擷取 (extract) 出影像中的物件或瑕疵區域，只要這些物件或瑕疵區域的灰階或顏色與背景有差異，我們就可以用門檻值法將這些物件或瑕疵上的點從背景分離出來

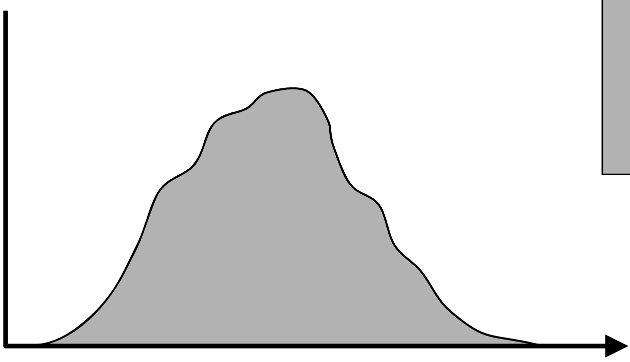
- 在自動化的應用上，門檻值演算法一定要能夠根據影像內容自行判定要用幾個門檻值，決定門檻值設在那裡；這是門檻值分割法的主要研究議題

- 某些影像的像素特徵會分成幾群，這些特徵的數量分佈圖就會呈現幾個波峰 (peak)，因此在判定要用幾個門檻值時，就可以很容易根據數量分佈圖的波峰數來決定，門檻值位置就設在波峰間的波谷 (valley) 處。

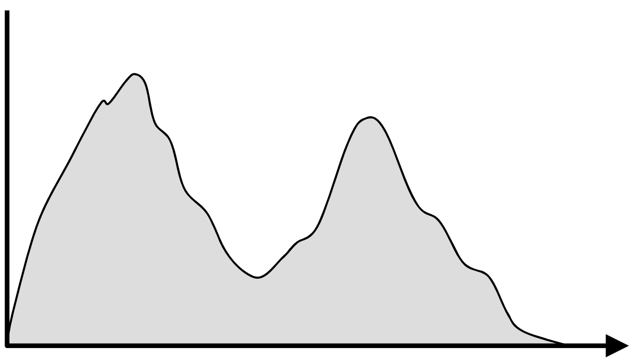
- 只有一個波峰的分佈圖稱為**單峰分佈圖 (unimodal histogram)**，有二個波峰的分佈圖稱為**雙峰分佈圖 (bimodal histogram)**，有多個波峰的分佈圖稱為**多峰分佈圖 (multimodal histogram)**。雙峰分佈圖用一個門檻值就夠分割了；多峰分佈圖有 *n* 個波峰，就要用 *n*-1 個門檻值才夠。

- 分佈圖類型

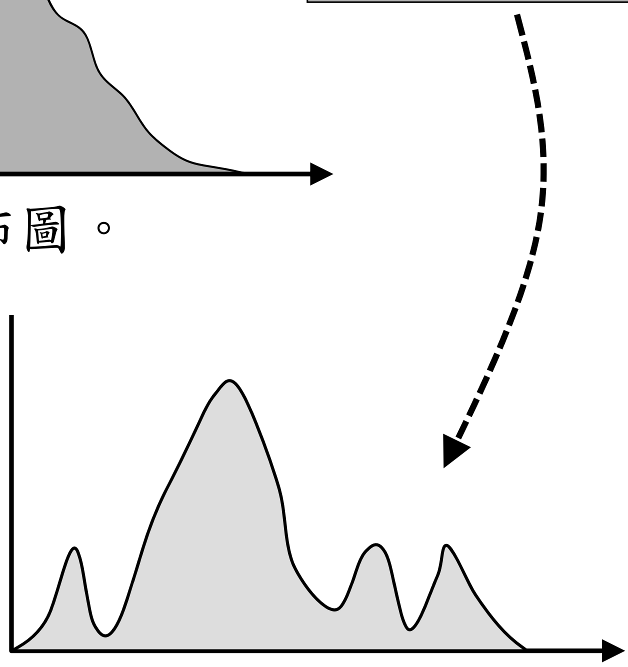




單峰分佈圖。



雙峰分佈圖。



多峰分佈圖。

9.1.1 門檻值法類別

- 門檻值法有三種類型：整體門檻值法、區域門檻值法、及動態門檻值法。這種分類完全依門檻值 (threshold value) 的選擇方法而定。
- **整體門檻值法** (global thresholding) 的門檻值選取只與影像灰階有關；例如，門檻值是從影像的灰階分佈圖找出來的；而且這個門檻值適用於整張影像的所有像素。做二值化則只選取一個門檻值；假設f是原始灰階，g是二值化後的數值，T是門檻值，則

$$g(x,y)=\begin{cases}1, & \text{if } f(x,y)>T\\ 0, & \text{if } f(x,y)\le T\end{cases}$$

- **區域門檻值法** (local thresholding) 的門檻值選取同時與影像灰階及各像素的區域特徵 (local property) 有關；例如，定義門檻值 T，

$$T=\sum_{i=-2}^{2}\sum_{j=-2}^{2}f(x+i,y+j)/25$$

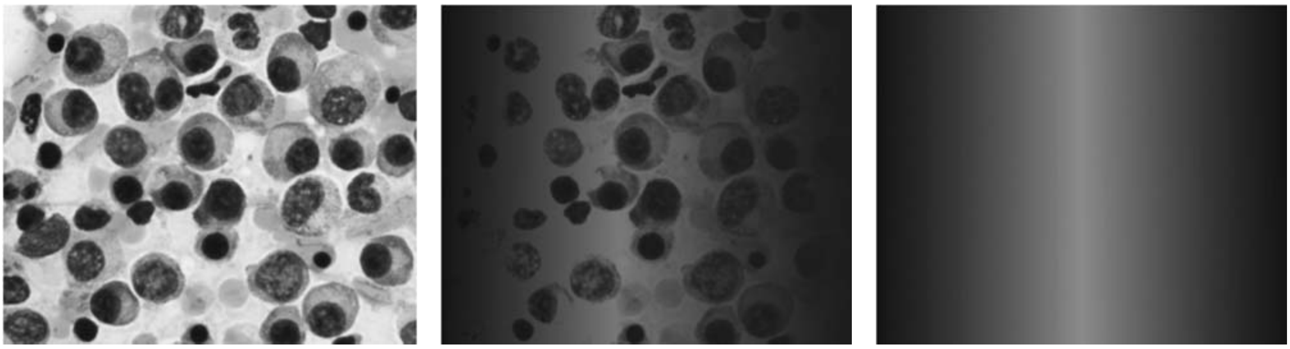
區域門檻值法還有另一個定義，就是先將影像分割成幾個重疊或不重疊區域；每一區域各別找一門檻值做整體門檻值分割。

- **動態門檻值法** (dynamic thresholding) 又稱為**適應性門檻值法** (adaptive thresholding)。動態門檻值法的門檻值選取同時與影像灰階、各像素的區域特徵、及像素位置 (location) 相關。

9.1.2 不均勻照度對門檻值法的影響

- 不均勻的環境光源照度對於門檻值法的影響很大

均勻光源的影像 $h(x, y)$　不均勻光源影像 $f(x, y)$　不均勻光源模式 $g(x, y)$



- 在相機固定拍攝位置的應用；例如，固定位置的生產線上平面物體之瑕疵檢測應用，門檻值法對於環境光源不均勻的問題是可以解決的。

- 步驟如下：

  S 1. 取一塊材質均勻的銀色或白色校正板放在相機前代替檢測物拍照，以取得校正照度用的影像 g。$g(x, y) = k\, i(x, y)$，其中 k 就是校正板的反射係數。

  S 2. 對於每一個待測物所拍攝的影像

  $f(x, y) = i(x, y)\, r(x, y)$.

  S 3. 將每一張 $f(x, y)$ 影像各別除以 $g(x, y)$ 影像，得到正規化 (normalized) 影像

  $h(x, y) = f(x, y) / g(x, y) = r(x, y) / k$。

  S 4. 將正規化 h 影像的灰階範圍線性放大到 [0, 255]，再對影像做門檻值分割；如此的門檻值法就不受不良環境照度的影響。

9.2 自動二值化

- 自動找出門檻值的方法稱為**自動門檻值法** (automatic thresholding)。若門檻值法只是將影像像素分成前景點與背景點，則自動門檻值法又稱為**自動二值化法** (automatic bi-level thresholding)。

- 本節將介紹下列 4 種自動二值化法方法：
  i. Otsu’s 群間最大差異二值化法
  ii. *Kapur* 的熵度量門檻值法 (entropy thresholding)
  iii. 最佳二值化法
  iv. 疊代逼近門檻值法 (Iterative thresholding)

9.2.1 群間最大差異二值化法

- 這個方法是由日本教授 Otsu 在1979年所提出的度量。假設一張影像的灰階尺度為 [0, 255]，灰階分佈圖函數為 $h(i)$, $i = 0, .., 255$，影像的像素個數為 N，i 灰階像素的機率值 (probability) 為 P_i，則

$$N = \sum_{i=0}^{255} h(i), \quad P_i = \frac{h(i)}{N}, \ i = 0, .., 255.$$

整個分佈圖函數的灰階平均值為 μ，變異數為 σ^2。

$$\mu = \sum_{i=0}^{255} i P_i, \quad \sigma^2 = \sum_{i=0}^{255} P_i (i-\mu)^2.$$

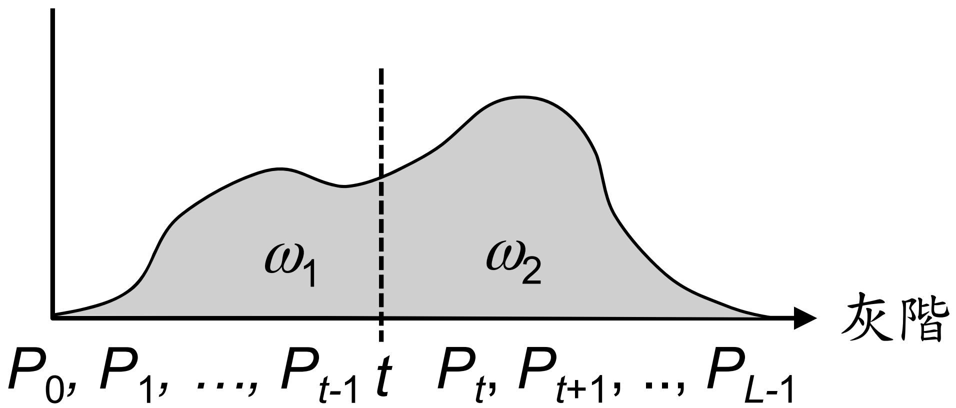
- 若以門檻值 t，將像素分成兩群。第一群的灰階範圍為 $[0, t-1]$，第二群的灰階範圍為 $[t, L-1]$；屬於第一群的機率值為 ω_1，屬於第二群的機率值為 ω_2；第一群的灰階平均值為 μ_1，變異數為 σ_1^2，第二群的灰階平均值為 μ_2，變異數為 σ_2^2。

群內的變異數 (with-in class variance) 為 σ_w^2，
群間的變異數 (between class variance) 為 σ_b^2，則

$$\omega_1 = \sum_{i=0}^{t-1} P_i, \quad \omega_2 = \sum_{i=t}^{L-1} P_i, \quad \mu_1 = \sum_{i=0}^{t-1} i\frac{P_i}{\omega_1}, \quad \mu_2 = \sum_{i=t}^{L-1} i\frac{P_i}{\omega_2},$$

$$\sigma_1^2 = \sum_{i=0}^{t-1} \frac{P_i}{\omega_1}(i-\mu_1)^2, \quad \sigma_2^2 = \sum_{i=t}^{L-1} \frac{P_i}{\omega_2}(i-\mu_2)^2,$$

$$\sigma_w^2 = \omega_1\sigma_1^2 + \omega_2\sigma_2^2, \quad \sigma_b^2 = \omega_1(\mu_1-\mu)^2 + \omega_2(\mu_2-\mu)^2.$$



$$\omega_1 = \sum_{i=0}^{t-1} P_i \;\Rightarrow\; \sum_{i=0}^{t-1} \frac{P_i}{\omega_1} = 1 \;\Rightarrow$$

$\dfrac{P_i}{\omega_1}$ = 在第一群中出現灰階 i 的機率值。

$$\mu_1 = \sum_{i=0}^{t-1} i\frac{P_i}{\omega_1}, \quad \sigma_1^2 = \sum_{i=0}^{t-1} \frac{P_i}{\omega_1}(i-\mu_1)^2.$$

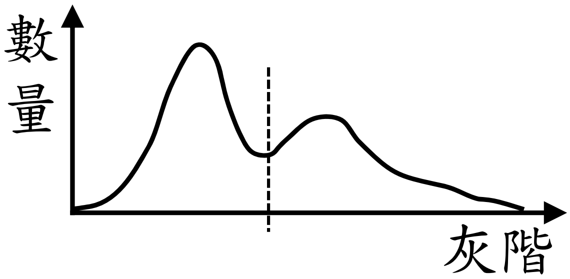
- *Otsu* 群間最大差異的評估準則定義為

  Maximize $f = \omega_1 (\mu_1 - \mu)^2 + \omega_2 (\mu_2 - \mu)^2$.

  準則的用法是以 1, 2, .., 254 等各數值當門檻值，各自算出一個 *f* 值，那一個門檻值算出來的 *f* 值最大，那個門檻值就是我們要的 *Otsu* 門檻值。

- *Otsu* 的群間最大差異法是要找有群間最大變異數的位置定義為門檻值。整張影像的灰階變異數值是固定的，且整張影像的灰階變異數值 σ^2 等於群內變異數 σ_w^2 加群間變異數 σ_b^2，

$$\sigma^2 = \sigma_w^2 + \sigma_b^2$$



- 所以找群間最大變異數相當於找最小的群內變異數值。*Otsu* 的評估準則就可改成為

  Minimize $f = \omega_1 \sigma_1^2 + \omega_2 \sigma_2^2$.

- 原始最大值的準則準則過於複雜，我們可以將它推導成更簡單的公式

  Maximize $f = \omega_1 \omega_2 (\mu_1 - \mu_2)^2$.

- ω_1 及 ω_2 的數值越相近，$\omega_1 \omega_2$ 的乘積就越大。μ_1 及 μ_2 的差異越大，$\mu_1 - \mu_2$ 就越大。所以 *Otsu* 的群間最大差異法就變成了要在兩群像素數量越相近，平均值差異越大的情形下，求出門檻值。

9.2.2 Kapur 的熵度量門檻值法

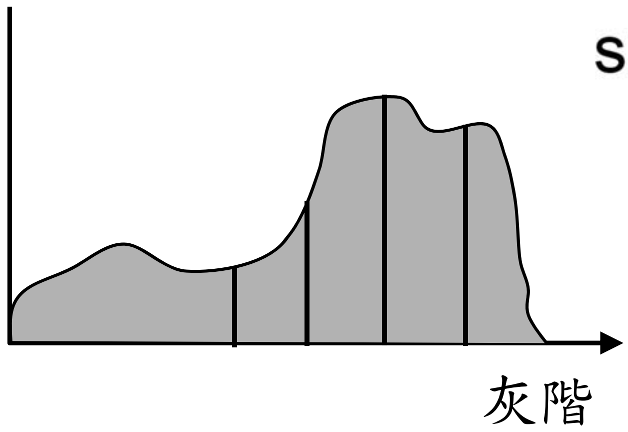
- 由 Kapur 等人在1985 年所提出的度量。假設影像的灰階尺度、灰階分佈函數、及灰階機率值的定義都與上一節一樣。熵度量法 (*entropy*) 的評估準則為

$$\text{Maximize } f = H_1 + H_2,$$

其中 $H_1 = -\sum_{i=0}^{t-1} \frac{P_i}{\omega_1}\log_2\frac{P_i}{\omega_1}, \quad H_2 = -\sum_{i=t}^{L-1} \frac{P_i}{\omega_2}\log_2\frac{P_i}{\omega_2}.$

- 一群像素的灰階分佈越均勻，則這一群像素的灰階熵值就越大；也就是各種灰階都有，每一種灰階的像素數都一樣，灰階熵值最大。所以熵度量法的意義是將像素分成兩群，希望各群像素灰階的分佈儘量為均勻分佈。

- 熵度量 (gray-level entropy) 的意義



$$\text{sum of probability} \ \sum_{i=1}^{8} p_i = ?$$

$$\text{gray-level } entropy = -\sum_{i=1}^{8} p_i \log_2 p_i$$

$$m = 8, \ p_1 = p_2 = \cdots = p_8 = \frac{1}{8} \ \Rightarrow \ H = -\sum_{k=1}^{8}\frac{1}{8}\log_2\frac{1}{8} = 3$$

$$m = 8, p_1 = 1, p_2 = 0, .., p_8 = 0 \ \Rightarrow \ H = -\sum_{k=1}^{8} p_i \log_2 p_i = 0$$

9.2.3 最佳二值化法

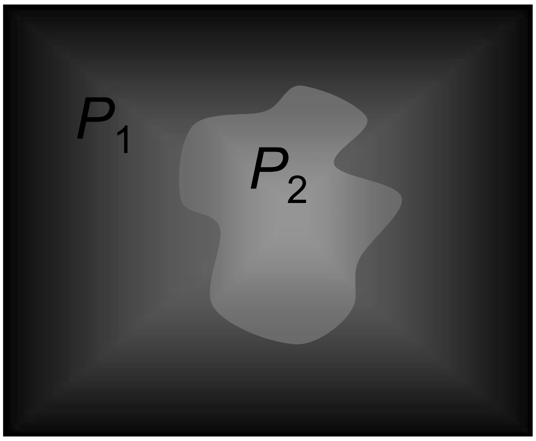
- 最佳二值化法 (Optimal thresholding) 是假設像素灰階具有某一種分佈，再定義一個誤差準則，在最小誤差的情形下找出二值化門檻值。最佳門檻值是根據數學的準則定義出來的，如果準則與真實問題有很大差異，即使數學算出來最好的答案，對於真實問題而言並不一定是最好的。
- 假設影像只有兩種灰階群，背景較暗，前景 (物體) 較亮。各群像素灰階為常態分佈 (normal distribution)。背景與前景的像素灰階一定有部份重疊；因此分群結果必然有部份像素分群錯誤。最佳二值化就是要在分群錯誤最小的情形下找出門檻值。

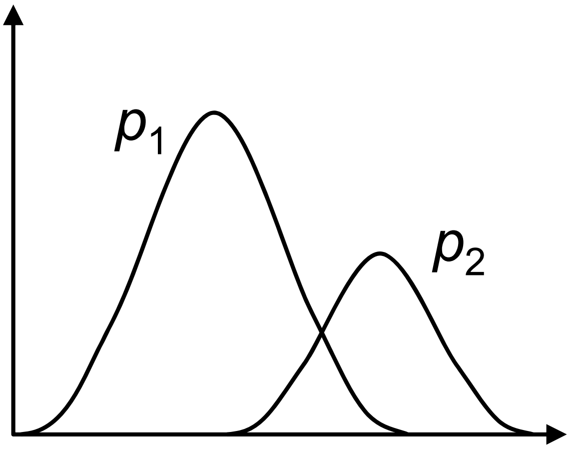
- 分群的錯誤率要用機率理論分析。假設背景與前景像素的事前機率 (prior probability) 各別為 P_1 及 P_2；則整張影像像素的複合機率密度函數 (mixture probability density function) 就定義為

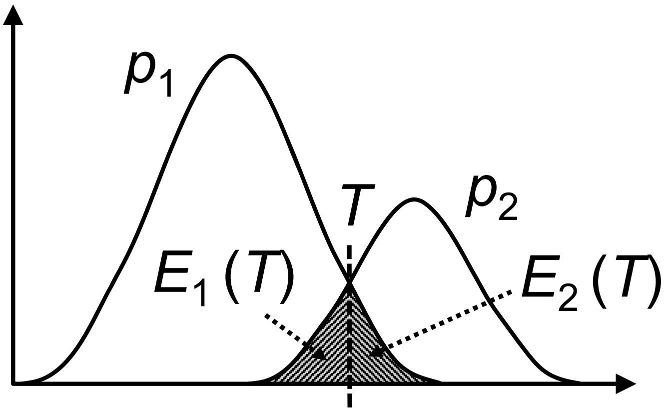
$$
\begin{aligned}
p(x) &= P_1 p_1(x) + P_2 p_2(x) \\
&= \frac{P_1}{\sqrt{2\pi}\,\sigma_1} exp\left[-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right] + \frac{P_2}{\sqrt{2\pi}\,\sigma_2} exp\left[-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right],
\end{aligned}
$$

其中μ_1 是背景點亮度的平均值，σ_1是背景點亮度的標準差，μ_2是前景點亮度的平均值，σ_2是前景點亮度的標準差；$P_1 + P_2 = 1$。

背景比前景暗，所以$\mu_1 < \mu_2$，二值化是要找一個門檻值 T，像素灰階 $< T$，則該像素將分類為背景點；否則分類為前景點。







在門檻值 T 左側的 p_2 函數值就是將前景點錯誤分類為背景點的機率值

$$E_1(T) = \int_{-\infty}^{T} p_2(x)dx.$$

在門檻值 T 右側的 p_1 函數值就是將背景點錯誤分類為前景點的機率值

$$E_2(T) = \int_{T}^{\infty} p_1(x)dx.$$

整張影像的像素分類錯誤機率值 (the overall probability of error) 則為

$$E(T) = P_2 E_1(T) + P_1 E_2(T).$$

最佳二值化就是要在誤差 $E(T)$ 最小的情形下求出門檻值 T。

- 求誤差函數 $E(T)$ 的極值 (極大值或極小值) 的方法很多，最簡單的方法就是用一次微分等於 0 的公式

$$\frac{d\,E(T)}{d\,T}=0$$

微分後的公式還非常複雜，我們只能借助微積分中的藍伯尼茲公式 (Liebnitz's rule) 導出

$$\frac{d\,E(T)}{d\,T}=0 \;\Rightarrow\; P_1 p_1(T)=P_2 p_2(T).$$

將常態分佈函數 $p_1(T)$ 及 $p_2(T)$ 都帶入上述公式，取對數 (logarithm) 後簡化成

$$AT^2+BT+C=0$$

其中

$$A=\sigma_1{}^2-\sigma_2{}^2,$$
$$B=2\left(\mu_1\sigma_2{}^2-\mu_2\sigma_1{}^2\right),\text{ and}$$
$$C=\sigma_1{}^2\mu_2{}^2-\sigma_2{}^2\mu_1{}^2+2\sigma_1{}^2\sigma_2{}^2\,ln\left[\sigma_2P_1/\sigma_1P_2\right].$$

根據二次式的解答

$$T=\frac{-B\pm\sqrt{B^2-4AC}}{2A}$$

有兩個解答，T 要介於 [1, 254] 之間，其中必有一個合理的解答。上述的解答公式過於複雜；我們可以另外看看兩個較簡單的特例

特例一：

如果前景與背景像素灰階具有相同的變異數

$$\sigma^2 = \sigma_1^2 = \sigma_2^2 \text{，則 } T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} ln\left[\frac{P_2}{P_1}\right].$$

特例二：

如果前景與背景像素灰階的事前機率相同, $P_2 = P_1$或變異數都等於 0, $\sigma_1 = \sigma_2 = 0$，則最佳門檻值即為

$$T = \frac{\mu_1 + \mu_2}{2}.$$

- 上述最佳二值化還有一個問題未解。在最佳二值化的解答中有五個參數數：μ_1 , σ_1, μ_2, σ_2, and P_1。所以要求出門檻值T，則這五個參數要先估計出來。

- 我們可以利用影像的灰階分佈圖 (gray-level histogram)，以最小平方誤差估計法 (least squares estimation) 估計這五個未知參數。以影像的實際灰階分佈圖及理論灰階分佈圖定義平方誤差 R

$$R = \frac{1}{256} \sum_{i=0}^{255} \left[p(x_i) - h(x_i)\right]^2$$

其中 h 是影像的實際灰階分佈圖，p是理論灰階分佈圖。R 是五個未知參數的函數。將R對這五個變數各別作偏微分並令為0，得五個方程式，

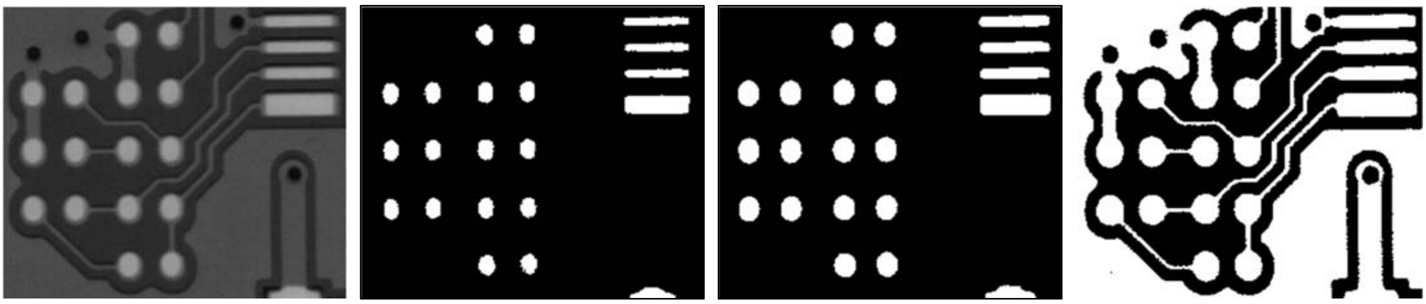
$$\begin{cases} \dfrac{\partial R}{\partial \mu_j} = 0, j = 1,2 \\ \dfrac{\partial R}{\partial \sigma_j} = 0, j = 1,2 \\ \dfrac{\partial R}{\partial P_1} = 0 \end{cases}$$

- 這五個方程式並不是線性方程式，我們無法用線性代數的聯立方程式 (system of linear equations) 求解答，而必須用複雜的方法求解答。過去有
  動態規劃法 (dynamic programming)、
  模擬退火法 (simulated annealing) 或
  基因演算法 (genetic algorithm) 被提出來解這個非線性聯立方程式的問題。

9.2.4 **疊代逼近門檻值**法 (*Iterative thresholding*)

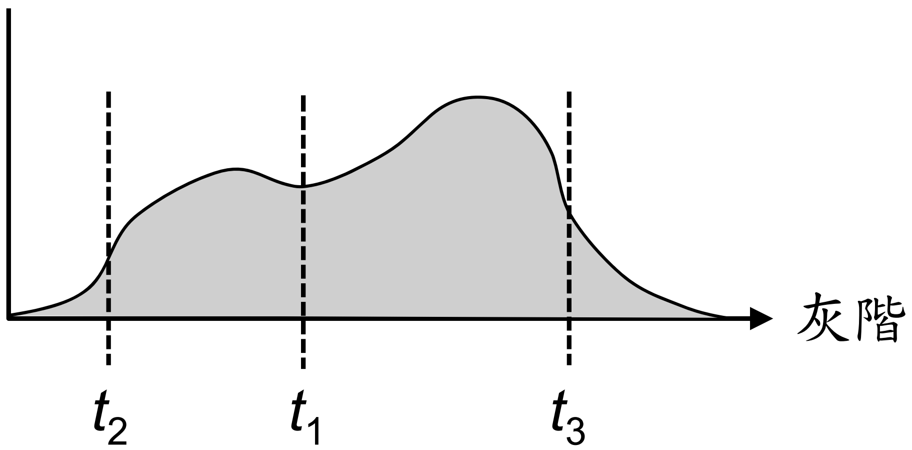
*S*1. 任意選取一個門檻值 (thresholding value)；例如，就以 128 做為起始門檻值。
*S*2. 根據門檻值，將像素分兩類。
*S*3. 如果疊代次數大於 10 次，則結束。
*S*4. 計算各群像素的 (灰階或邊界強度) 平均值 μ_1, μ_2
*S*5. 設定 $\mu = (\mu_1 + \mu_2) / 2$ 為新的門檻值；再回到 *S*2

- 疊代逼近門檻值法的效率比前述三種方法好；但疊代法僅適合黑白較分明的影像；例如 *PCB* image。



9.3 自動多值化

- 許多自動二值化法都可擴充成自動多值化法。理論上多值化的門檻值是彼此相互影響的；所以最好的方法是多個門檻值一起求出來；例如，Otsu 多值化準則 minimize $\omega_1 \sigma_1^2 + \omega_2 \sigma_2^2 + \omega_3 \sigma_3^2$
但就是太耗時了；例如，*L* 灰階影像要找 *n* 個門檻值，則第一個門檻值有*L*-2 個選擇，第二個門檻值有 *L*-3 個選擇，其餘依次推；求 *n* 個門檻值總共要測試 (*L*-2) (*L*-3) .. (*L*-*n*-1) 種組合；計算複雜度將從 *O*(*L*) 急速升高至 $O(L^n)$；而且要做多少值化，要使用者先行決定，演算法不能根據影像的灰階分佈來決定要設幾個門檻值。

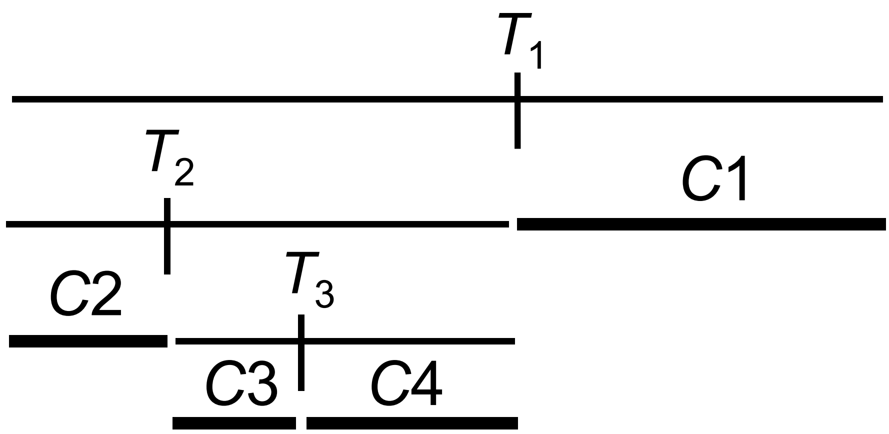


方法1. 所有門檻值同時一起找

方法2. 所有門檻值循序一個一個找

- 所以現在常用的多值化法，大多是循序一個一個求出；再另外訂定一個準則，來決定該影像需要幾個門檻值。如此方法要找出 n 個門檻值，所需要的運算次數只要 $(L\text{-}2) + (L\text{-}3) + .. + (L\text{-}n\text{-}1)$；計算複雜度只有 $O(nL)$。
- 雖然循序疊代法所求出的結果只是近乎最佳解而已，不過已經相當不錯了。

- *Otsu* 自動多值化法
  *S*1. 執行 *Otsu* 自動二值化，將像素分二群。
  S2. 各群各自分析灰階變異數 (gray-level variance)；若某一群的變異數大於門檻值 *T* (不完整)，則將該群像素帶回 *S*1 (再執行 Otsu自動二值化)。若某一群的變異數小於等於門檻值 *T*，則該群像素就停止繼續二值化。



9.4 區塊分割法

- 區塊分割 (region segmentation) 的目的是要將影像分割成各種不同特性的區域；每一區域中的像素在影像中是相連且特性相似。
- 上述的特性包括灰階、色彩、紋理、.. 等。區塊分割與前述分佈圖門檻值分割 (thresholding) 的最大不同在於區塊分割不僅考慮灰階或色彩值，還同時考慮像素的空間連續性 (spatial continuity)。
- 區塊分割法最重視像素在空間的相連性，因此大部份區塊分割法都是從空間相連的關係開始構思分割演算法。本章將介紹區域分割法中最常使用的區域成長法 (region growing) 及區域分割與合併 (splitting and merging)。

9.4.1 區塊分割的基本定義

- 假設影像*R*被區塊分割法分割成 *n* 個區塊 (subregions)：$R_1, R_2, ..., R_n$，則區塊分割法的分割結果必須要滿足下列五個條件：
  - *i.* 整張影像的每一像素都必須被分區，$\bigcup_{i=1}^{n} R_i = R$
  - *ii.* 每一區塊必須是一個完整的連結區塊 (R_i is a connected region, i = 1, 2, ..., n.)
  - *iii.* 沒有任一像素可以被分類到兩不同區塊，$R_i \cap R_j = \phi$ for all i and j, $i \neq j$.
  - *iv.* 同一區塊內的像素都具備相似的特性，$P(R_i) = True$, for i = 1, 2, ..., n。
  - *v.* 不同特性的像素一定分到不同的相鄰區塊，$P(R_i \cup R_j) = False$, for $i \neq j$, i, j are connected。

9.4.2 區域成長法

- 區域成長 (region growing by pixel aggregation) 是一個區域分割的最簡單方法，從一個點的小區域逐漸擴大範圍；區域擴大的原則是與區塊相鄰且特性相似的像素就會被歸屬成這個區域的一份子。區域擴大的方式可以疊代 (iterative) 的分析已分類過之像素的3×3或5×5遮罩範圍內之像素；分析的方式是比較這些像素與起始點像素的灰階、色彩值、或其他特性。當它們的特性差不多時，這些像素就被分類成與起始點同一個區域。

- 一般化的演算法如下

  *S*1. 從影像中選取一個未處理過的像素當作起始點 (initial seed)，並定義一個新區塊。

  *S*2. 根據事前所定義的特性，比較已處理點之鄰點 (neighboring pixels) 與起始點的相似度 (similarity)；如果夠相似，則將該鄰點併入目前這個區塊。重複執行 *S*2，處理鄰點的鄰點；一直到沒有鄰點為止。

  *S*3. 若還有未處理過的像素，則回到*S*1；否則停止。

- 範例

	1	2	3	4	5
1	0	0	5	6	7
2	1	1	5	8	7
3	0	**1**	6	**7**	7
4	2	0	7	6	6
5	0	1	5	6	5

(a) 原始影像。

a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b

(b) $|x_i - x_j| < 3$.

a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a

(c) $|x_i - x_j| < 8$.

- 區域成長的過程中，最重要的是選擇像素之相似度的評估準則；準則不一樣，成長出來的區域就不一樣。

- 在區域成長中有三件事情要考慮，
  第一個就是選擇一個起始點，
  其次是選擇相似性的度量，
  第三是在區域成長中，成長到那種階段要停下來？
  也就是
  i. 如何選取起始點？
  ii. 可以使用什麼特性評估相似？
  iii. 何時停止區域擴張 (成長)？

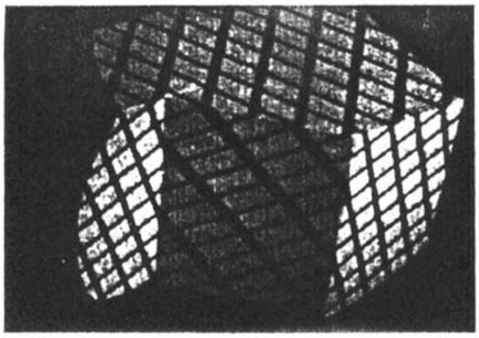
A. 如何選取起始點

- 就區塊分割的基本定義之第一條件而言，整張影像的每一像素都必須被分區；那麼以哪一像素當起始點都可以，只是結果可能有少許不一樣。
- 另外，某些應用可以用具有特殊意義的像素當起始點；例如，用紅外線影像 (infrared image) 偵測夜間海面上的船艦。船艦有馬達在運轉，溫度較高，馬達部位在紅外線影像中較亮；那麼我們就可以定義多少灰階以上的像素都可拿來當起始點。

B. 可以使用什麼特性評估相似

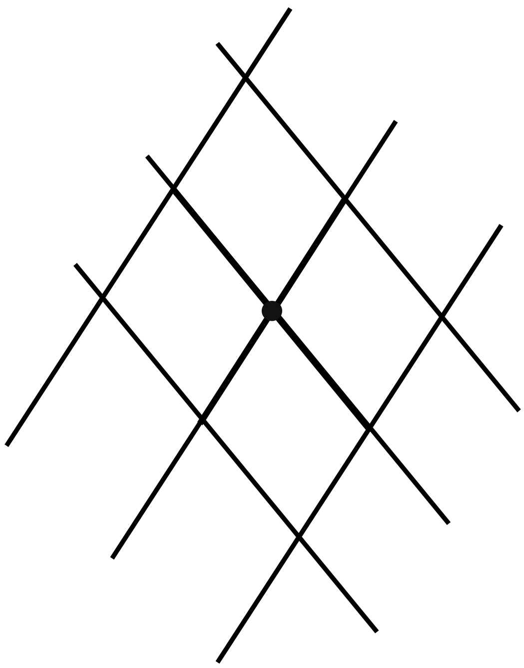
- 區域成長法中，最重要的是像素相似度的評估準則。一般常用的特徵有：灰階、色彩值、或紋理。比較各種特徵之相似程度時，要與起始點做比較，不是與相鄰點做比較。
- 用紋理做區塊成長，紋理可以有微觀與巨觀的不同定義。

- 以下是一個巨觀紋理的分割應用。原始影像是一個表面有平行雷射網格光影的多面體影像。物體表面都是平面，雷射光與物體表面法線 (normal vector) 的夾角不同，因而在物體表面上形成不同夾角的平行線條光影。
- 每一種平行線條光影代表一個平面；我們的最終目的是要從這一張影像求出每一個平面的3D線性方程式。要達到這個目的，第一步驟就是要把所有平面分割開來。

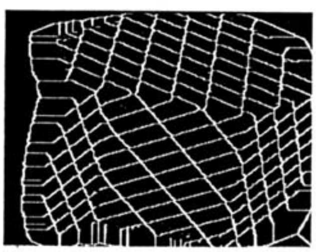


- 每一平面上的光影都是一個交叉點與四個交叉點相鄰，我們可以用格子形狀來分割區塊；
- 格子分群步驟

*S*1. 影像反白再做二值化，



*S*2. 細線化，



*S*3. 找出所有交叉點，並只保留有四相鄰點的交叉點，

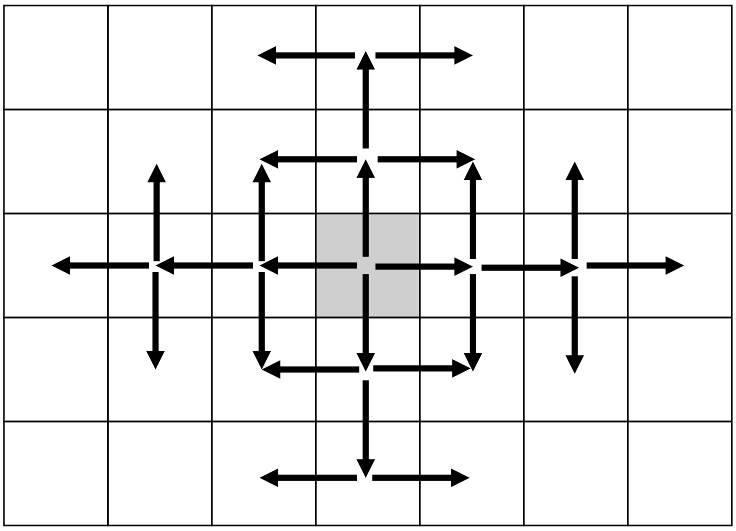
*S*4. 每一個完整的交叉點都有四個鄰點，以四個鄰點的方向與長度當做中央交叉點的特徵向量，以這個特徵向量做為相鄰交叉點的相似度度量，以進行區域擴張。

C. 何時停止區域擴張 (成長)

- 一般做法是，一個區塊的周邊已經沒有點可以合併就停止區域成長；影像中所有點都已分群了，整個區域成長就結束。有些應用會以成長區域的形狀 (外型) 或面積大小定義區域成長停止準則 (stopping criterion)；只是區域形狀不易定義，而且準確度不高。

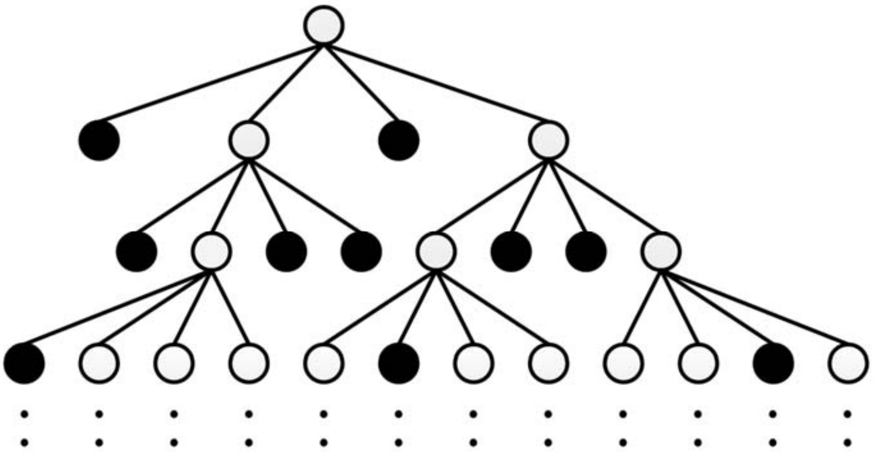
D. 區域擴張次序

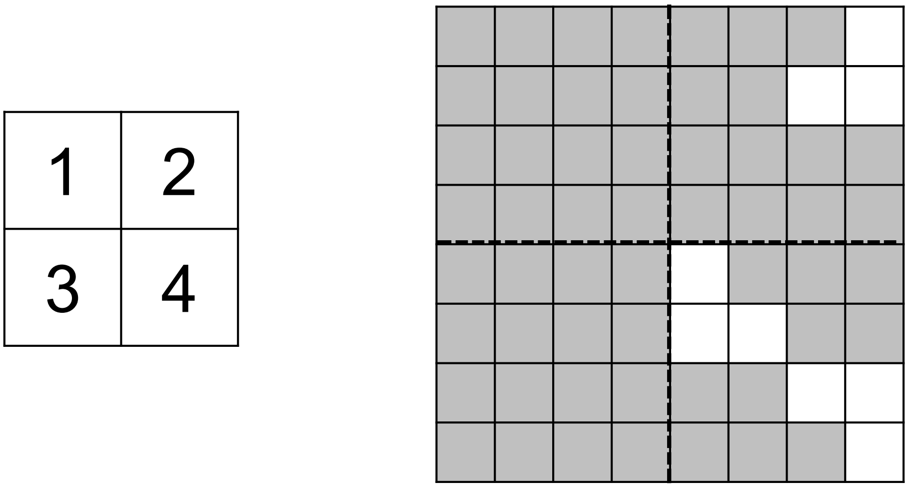
- 從起始點往上下左右四個方向擴張的模式



9.4.3 區域分割合併法

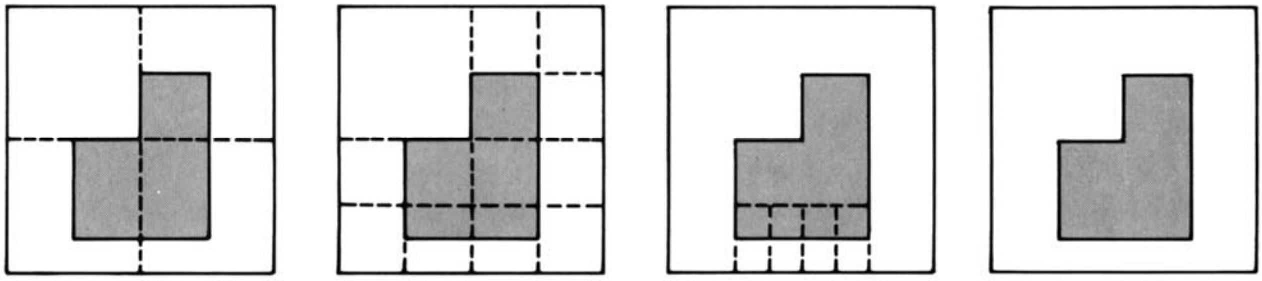
- 區域分割合併法 (region splitting and merging) 是分割與合併兩種處理輪流交替使用；區域成長法只是一直做合併而以，所以分割合併法在使用上比區域成長法更有彈性。另外還有一點不一樣的是，區域成長法是一個個像素各別考慮合併，但分割合併法是一個個區域各別考慮合併。
- 為了讓分割有效率，這裡採用四分樹 (quadtree) 資料結構。





- 等分割區塊內的像素屬性不一樣時 (也就是前面理論所講的，邏輯述語 $P(R_i) = False$)，則該區塊將一定再做一樣的等分割。做完一次分割，即執行合併。只要是相鄰且屬性相同的區塊，不管大小都將一次合併成一個大區塊。分割與合併交替執行，直到不能再分割及不能再合併為止。
- 其演算法流程如下所示
  - $S1.$ 將所有滿足 $P(R_i) = False$ 的區塊 R_i 等區域四分割
  - $S2.$ 將所有滿足 $P(R_i \cup R_j) = True$ 的相鄰區塊 R_i 及 R_j 合併成一大區塊
  - $S3.$ 若還有區塊 R_i 滿足 $P(R_i) = False$ 則回到 $S1$，否則停止。

簡單範例

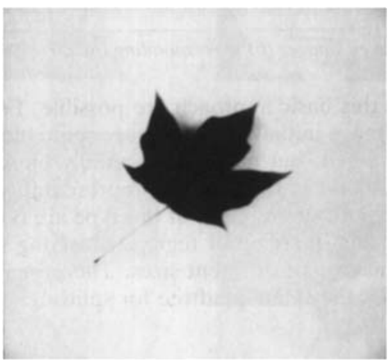
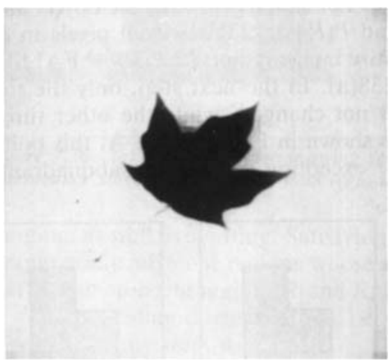
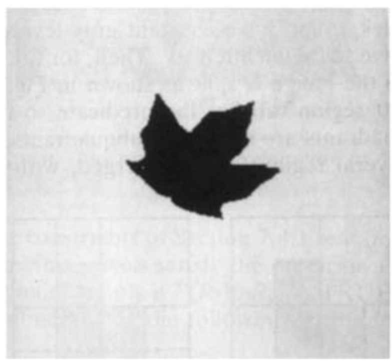


(a) (b) (c) (d)

- 對於一般影像於言，不太容易達到一個區塊內的所有像素屬性都一樣；因此對於相同屬性的定義(也就是$P(R_i) = True$) 可以放鬆一點；例如，當一區塊中有 90 % 像素滿足 $\left|f(x,y)-\bar{f}\right| < 3\sigma$ ，即定義為 $P(R_i) = True$；其中 $\bar{f}$ 及 σ 各別是該區塊內像素灰階平均及變異數。

範例。

當一區塊中有 80 % 像素滿足 $\left|f(x,y)-\bar{f}\right| < 2\sigma$ ，即定義 $P(R_i) = True$。當區塊 R_i 的 $P(R_i) = True$，則該區塊所有像素的灰階即改變為 $\bar{f}$。

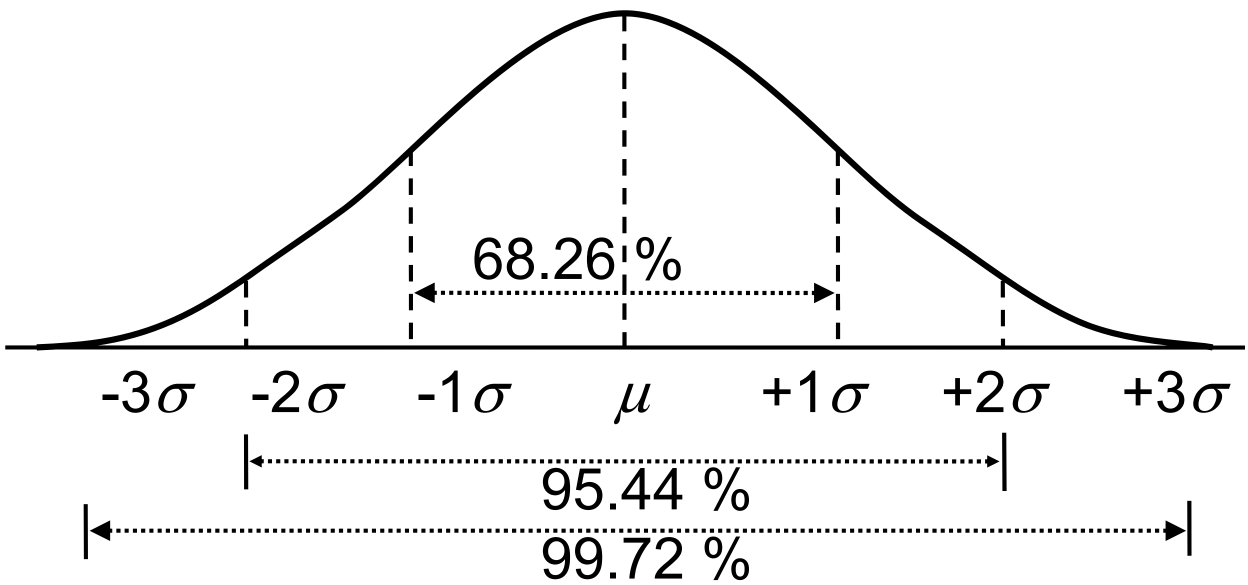




原始影像。 合併分割過程。 最後結果。

- 在決策判定 (decision making) 中，我們常會假設資料分佈為常態分佈 (normal (Gaussian) distribution)，然後使用準則 (criterion)

$$|x - \mu| \leq \lambda\, \sigma,$$

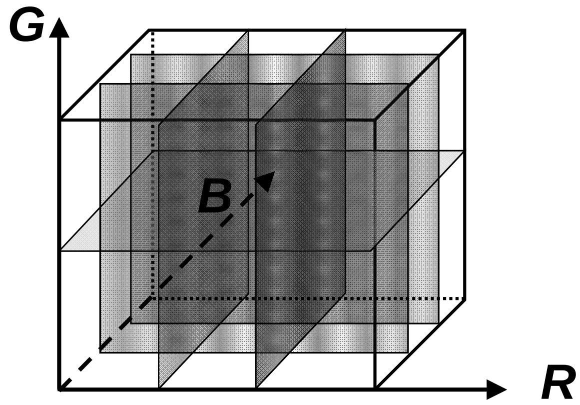
其中 λ = 1, 2, 3, ..，判定結果



9.5 彩色影像分割

- 所有灰階影像分割法都可用在彩色影像上；另外，彩色影像相當於三頻譜的灰階影像，因此彩色影像的分割還可以在三度色彩空間上處理。不過一般自然物體；例如，膚色、樹幹、花瓣、岩石、路面、 .. 等，不可能是均勻的色度；因此彩色影像分割難度相當高。

- 對於彩色多波段或多頻譜影像的分割方法大致有兩大類：一是門檻值分割法，另一是多頻譜空間群聚法 (clustering)。
- 以 *RGB* 色彩為例，門檻值分割法可以*R*, *G*, *B* 三波段各自獨立做多值化，最後再將結果交集出來。每一區塊代表一個影像中的彩色像素群。



- 多頻譜空間群聚法就是直接在多頻譜空間 (或色彩空間) 中，就多頻譜資料 (或彩色資料) 做群聚 (clustering)；群聚的做法其結果比門檻值法正確，但執行時間也比門檻值法耗時許多。
- *K*-means 群聚法

  S_1. 隨機選取 k 個資料當作初始群中心 $\mu_1, \mu_2, .., \mu_k$ 。

  S_2. 以最短距離準則 $\min_j |x_i - \mu_j|$，將每個資料 x_i 分到對應的群 j。

  S_3. 重新計算群中心 μ_j；若所有群中心變異都小於門檻值，則停止；否則回到 S_2.

- 彩色影像分割可以直接在 *RGB* 色彩空間上做，也有許多方法是將色彩先轉到 *IHS*, *VHS*, *Yuv*, *YCbCr*, .. 等色彩空間再分割。理論上所有色彩空間都差不多，只是就各別應用不同，色彩群聚 (clustering) 效果會有一點不一樣而已。之所以會先做上述色彩轉換，通常都是為了分割結果不要受到光線強弱或陰影因素的影響。這時我們先將顏色分成亮度與色度，再針對亮度及色度各別分割或處理。